Julian Diaz
Mr.Orkney
Period 3
2-15-18

Written Response

2a. The programming language that I used is called Python. I chose this program because I think explaining the coding was easier and when I was typing my code it was easy to have it working in my opinion. The purpose of my program is to have you play a tic-tac-toe game against a robot. I decided to have someone play against a robot instead of having that person play against another person because if you play against the system it is more challenging but the fun part of the game is to mainly rage and lose more than win. The video that I made illustrates how the game works from my perspective and it will as well have the code included in there. It would also show what type of responses return when I respond to the questions that are being asked to do.

2b. The process of making my program was quite difficult because I started off doing it individually and not knowing what each piece of code meant but at end I had to get many help from my partners so they can explain to me what is the purpose to each line. Throughout the process of making my game, I was getting stuck on some parts because on my game the board was too big for the “x” and the “o” so I wanted to change that . I solved that problem by deleting many lines that were in the coding and then I figured out which lines do I need to keep and which ones don’t I need so I am able to have the proper size board. Another thing that was difficult during making my game was having to spell every line of code in a specific way, in order for the program to run it. I figured that out by trying different things and retyping lines in order to see if the program allows it to happen without any errors. It did take some time but what I had to do was research and figure out what do I have to enter for the first part of the line.

2c. Algorithm:

```
for i in range(1, 10):
copy = getBoard(board)
if isSpaceFree(copy, i):
makeMove(copy, computerLetter, i)
if isWinner(copy, computerLetter):
        return i

for i in range(1, 10):
copy = getBoard(board)
```

```
if isSpaceFree(copy, i):
makeMove(copy, playerLetter, i)
if isWinner(copy, playerLetter):
    return i
```

The purpose of my algorithms is to let the computer or player enter their numbers that were chosen from both of them onto a board and everytime they type a new number it’s copied onto a new board with the their numbers that they’ve already chosen. It also checks if the computer or player can win in the next move by typing the next number. The algorithm that is in my program has a 2 part to it. When I say that it means that there is an algorithm for the computer and for the player because it has to tell the computer that the number range is to 1 to 10 and for the player it has the same range as well. Each algorithm isn’t really different except that one is for the computer and one is for the player but how both algorithm functions work is that whether it is the computers or the players turn it lets them choose a number from 1 to 10 and then after they’ve chosen a number then it is inserted onto the board. Then when it’s the computers turn then it returns another board with the letter that chosen from the other person.

2d.

```
Abstraction: def getPlayerMove(board):
        move = ' '
        while move not in '1 2 3 4 5 6 7 8 9'.split() or not isSpaceFree(board, int(move)):
        print('What is your next move? (1-9)')
        move = input()
        return int(move)

        def getComputerMove(board, computerLetter):
        # Given a board and the computer's letter, determine where to move and return
        that move.
        if computerLetter == 'X':
        playerLetter = 'O'
        Else:
        playerLetter = 'X'
```

The abstraction kept the program moving and by that I mean that it didn’t ruin it or have difficulties during the process of playing the game. It helped because it would keep asking the question which is “what is your next move” and they are only able to choose numbers that are in the number scale of 1 to 9. Another reason that it helped was that it didn’t just stop asking “what is your next move”, it just kept asking until either the computer or player won. The computers part of the coding as you can see is not the same as the players code and that is because you don’t have control over the computer so the computer can only choose the number on it’s own and then it is returned onto the board. Another reason that it helped was that it didn’t just stop asking “what is your next move”, it just kept asking until either the computer or player won. This part did include a list of numbers so that means it had a mathematical concept to it and the logical concept part is that it asks a question repeatedly with a number scale telling you what do you want to do next.